

## CLAIMS

We claim:

1           1.     A method in a computer system for executing applications to  
2     communicate with remote shared service providers using multiple messaging models, each  
3     remote shared service provider having for each of the multiple messaging models a distinct  
4     associated proxy component able to communicate with that remote shared service provider  
5     using that messaging model, the method comprising:  
6                 under control of a first of the executing applications having a local messaging  
7     component able to communicate with each of the associated proxy components,  
8                 notifying the local messaging component to send a specified first message  
9     to a specified first remote shared service provider using a specified first messaging model;  
10     and  
11                 under control of the local messaging component, sending the first message  
12     to the first remote shared service provider using the first messaging model by,  
13                 retrieving configuration information for the first remote shared  
14     service provider that identifies a first proxy component that is associated with the first remote  
15     shared service provider and is able to communicate with the first remote shared service  
16     provider using the first messaging model;  
17                 when the first proxy component is not already instantiated within the  
18     first application, instantiating the first proxy component;  
19                 sending a message to the first proxy component requesting that the  
20     first proxy component send the first message to the first remote shared service provider using  
21     the first messaging model; and  
22                 receiving a response from the first remote shared service provider via  
23     the first proxy component, and notifying the first application of the received response,  
24     so that an application can communicate with any of the remote shared service providers  
25     using any of the multiple messaging models via a local messaging component that identifies

26 and uses a proxy component able to communicate with a specified remote shared service  
27 provider using a specified messaging model.

1           2.     The method of claim 1 including, under control of the first application,  
2 communicating with the first remote shared service provider using a second messaging model  
3 by:

4                 notifying the local messaging component to send a specified second  
5 message to the first remote shared service provider using a specified second messaging  
6 model; and

7                 under control of the local messaging component, sending the second  
8 message to the first remote shared service provider using the second messaging model by,

9                     retrieving configuration information for the first remote shared  
10 service provider that identifies a second proxy component that is associated with the first  
11 remote shared service provider and is able to communicate with the first remote shared  
12 service provider using the second messaging model, the second proxy component distinct  
13 from the first proxy component;

14                     when the second proxy component is not already instantiated within  
15 the first application, instantiating the second proxy component;

16                     sending a message to the second proxy component requesting that  
17 the second proxy component send the second message to the first remote shared service  
18 provider using the second messaging model; and

19                     receiving a response from the first remote shared service provider via  
20 the second proxy component, and notifying the first application of the received response.

1           3.     The method of claim 1 including, under control of the first application,  
2 communicating with a second of the remote shared service providers using the first  
3 messaging model by:

4                 notifying the local messaging component to send a specified second  
5 message to a specified second remote shared service provider using the first messaging  
6 model; and

7 under control of the local messaging component, sending the second  
8 message to the second remote shared service provider using the first messaging model by,  
9 retrieving configuration information for the second remote shared  
10 service provider that identifies a second proxy component that is associated with the second  
11 remote shared service provider and is able to communicate with the second remote shared  
12 service provider using the first messaging model, the second proxy component distinct from  
13 the first proxy component;

14 when the second proxy component is not already instantiated within  
15 the first application, instantiating the second proxy component;

16 sending a message to the second proxy component requesting that  
17 the second proxy component send the second message to the second remote shared service  
18 provider using the first messaging model; and

19 receiving a response from the second remote shared service provider  
20 via the second proxy component, and notifying the first application of the received response.

1 4. The method of claim 1 including:

2 under control of a second of the executing applications having a second local  
3 messaging component able to communicate with each of the associated proxy components,

4 notifying the second local messaging component to send a specified  
5 second message to the first remote shared service provider using the first messaging model;  
6 and

7 under control of the second local messaging component, sending the  
8 second message to the first remote shared service provider using the first messaging model  
9 by,

10 retrieving the configuration information for the first remote shared  
11 service provider that identifies the first proxy component;

12 instantiating the first proxy component;

13 sending a message to the first proxy component requesting that the  
14 first proxy component send the second message to the first remote shared service provider  
15 using the first messaging model; and

16 receiving a response from the first remote shared service provider via  
17 the first proxy component, and notifying the second application of the received response.

1 5. The method of claim 1 including, under control of the first proxy  
2 component:

3 receiving the sent message from the local messaging component;  
4 identifying based on the received message that the first messaging model is to  
5 be used for communicating;  
6 sending the message to the first remote shared service provider;  
7 receiving a response from the first remote shared service provider; and  
8 sending the received response to the local messaging component using the first  
9 messaging model.

1 6. The method of claim 5 including, under control of the first remote  
2 shared service provider:

3 receiving the message sent from the first proxy component;  
4 performing processing associated with the received message, the processing  
5 generating a response; and  
6 sending the generated response to the first proxy component.

1 7. The method of claim 1 wherein the specifying of the first remote shared  
2 service provider in the notifying of the local messaging component includes a unique name  
3 for the first remote shared service provider, and wherein the retrieving of the configuration  
4 information for the first remote shared service provider is based at least in part on the  
5 included name.

1 8. The method of claim 1 wherein the specified first message is a request  
2 for one of multiple distinct services provided by the first remote shared service provider,  
3 wherein the notifying of the local messaging component further identifies a name of the one

4 service, and wherein the identifying of the first proxy component from the retrieved  
5 configuration information is based at least in part on the identified name of the one service.

1 9. The method of claim 1 wherein the first proxy component is a software  
2 module, and wherein the instantiating of the first proxy component includes executing the  
3 software module.

1 10. The method of claim 1 wherein the first proxy component is an instance  
2 of an object class, and wherein the instantiating of the first proxy component includes  
3 creating the instance.

1 11. The method of claim 10 wherein the sending of the message to the first  
2 proxy component includes invoking an interface method of the first proxy component  
3 instance.

1 12. The method of claim 11 wherein the first proxy component instance has  
2 distinct interface methods corresponding to each of the messaging models with which the  
3 first proxy component is able to communicate, and wherein the request that the first proxy  
4 component use the first messaging model is based on selecting the interface method of the  
5 first proxy component that corresponds to the first messaging model as the interface method  
6 to be invoked.

1 13. The method of claim 11 wherein the request that the first proxy  
2 component use the first messaging model is based on specifying a parameter during the  
3 invoking of the interface method such that the specified parameter corresponds to the first  
4 messaging model.

1 14. The method of claim 1 wherein the local messaging component is an  
2 instance of an object class.

1           15.     The method of claim 14 wherein the notifying of the local messaging  
2 component includes invoking an interface method of the local messaging component  
3 instance.

1           16.     The method of claim 15 wherein the local messaging component  
2 instance has distinct interface methods corresponding to each of the messaging models with  
3 which the local messaging component is able to communicate, and wherein the specifying of  
4 the first messaging model during the notifying of the local messaging component is based on  
5 selecting the interface method of the local messaging component that corresponds to the first  
6 messaging model as the interface method to be invoked.

1           17.     The method of claim 15 wherein the specifying of the first messaging  
2 model during the notifying of the local messaging component is based on specifying a  
3 parameter during the invoking of the interface method such that the specified parameter  
4 corresponds to the first messaging model.

1           18.     The method of claim 1 wherein the configuration information is  
2 retrieved from storage via a directory service using a unique name of the first remote shared  
3 service provider.

1           19.     The method of claim 18 including, before the executing of the first  
2 application, storing the configuration information using the directory service.

1           20.     The method of claim 1 wherein the retrieved configuration information  
2 includes initialization information for the first proxy component, and wherein the  
3 instantiating of the first proxy component includes providing the initialization information to  
4 the first proxy component for use as part of its initialization.

1           21.     The method of claim 1 wherein the message sent to the first proxy  
2 component is a version of the specified first message that is transformed under the control of  
3 the local messaging component before the sending.

1           22.     The method of claim 1 wherein the first application includes multiple  
2 action handler components each for responding to requests from external clients with  
3 responses generated in a format used by the action handler component, and wherein the  
4 notifying of the local messaging component to send the specified first message is prompted  
5 by a request generated by one of the action handler components.

1           23.     The method of claim 22 wherein the first application includes multiple  
2 view handler components each for transforming generated responses in formats used by  
3 action handler components into formats compatible with external clients, and including, after  
4 the notifying of the received response from the first remote shared service provider, using  
5 one of the view handler components to transform the received response into a format  
6 compatible with an external client.

1           24.     The method of claim 1 wherein the first messaging model is  
2 synchronous communication.

1           25.     The method of claim 1 wherein the first messaging model is  
2 asynchronous communication.

1           26.     The method of claim 1 wherein the first messaging model is request-  
2 reply communication.

1           27.     The method of claim 1 wherein the first messaging model is one-way  
2 communication.

- 1 28. The method of claim 1 wherein the first messaging model is store-and-  
2 forward communication.
- 1 29. The method of claim 1 wherein the first messaging model is queued  
2 communication.
- 1 30. The method of claim 1 wherein the first messaging model is publish-  
2 subscribe communication.
- 1 31. The method of claim 1 wherein the first messaging model is broadcast  
2 communication.
- 1 32. The method of claim 1 wherein the first messaging model is  
2 conversational communication.
- 1 33. The method of claim 1 wherein the first remote shared service provider  
2 does not provide a response that corresponds to the first messaging model, and wherein the  
3 first proxy component generates the response.
- 1 34. The method of claim 1 wherein the received response from the first  
2 remote shared service provider does not correspond to the first messaging model, and  
3 wherein the local messaging component generates a response that corresponds to the first  
4 messaging model to be used for the notifying of the first application.



1           35. A computer-readable medium whose contents cause a computing device  
2 to execute applications able to communicate with remote shared service providers using  
3 multiple messaging models, each remote shared service provider having for each of the  
4 multiple messaging models a distinct associated proxy component able to communicate using  
5 that messaging model, by:

6           under control of a first of the executing applications having a local messaging  
7 component able to communicate with each of the associated proxy components,

8           notifying the local messaging component to send a specified first message  
9 to a specified first remote shared service provider using a specified first messaging model;  
10 and

11           under control of the local messaging component, sending the first message  
12 to the first remote shared service provider using the first messaging model by,

13           retrieving configuration information for the first remote shared  
14 service provider that identifies a first proxy component that is associated with the first remote  
15 shared service provider and is able to communicate using the first messaging model;

16           when the first proxy component is not already instantiated,  
17 instantiating the first proxy component; and

18           sending a message to the first proxy component requesting that the  
19 first messaging model be used by the first proxy component in sending the first message to  
20 the first remote shared service provider.

1           36. A computing device for executing applications to communicate with  
2 remote shared service providers using multiple messaging models, each remote shared  
3 service provider having for each of the multiple messaging models a distinct associated proxy  
4 component able to communicate with that remote shared service provider using that  
5 messaging model, comprising:

6           a first application capable of, during execution, notifying a local messaging  
7 component able to communicate with each of the associated proxies to send a specified first

8 message to a specified first remote shared service provider using a specified first messaging  
9 model; and

10 the local messaging component capable of, during execution, retrieving  
11 configuration information for the first remote shared service provider that identifies a first  
12 proxy component that is associated with the first remote shared service provider and is able  
13 to communicate with the first remote shared service provider using the first messaging  
14 model, instantiating the first proxy component, and sending a message to the first proxy  
15 component requesting that the first proxy component send the first message to the first  
16 remote shared service provider using the first messaging model.

1 37. A method in a computer system for an executing application to  
2 communicate with remote services using multiple messaging models, each remote service  
3 having for each of the multiple messaging models an associated proxy able to communicate  
4 using that messaging model, the method comprising:

5 for each of multiple of the remote services,

6 notifying a local messaging service to send a message to the remote  
7 service using a specified messaging model;

8 under control of the local messaging service,

9 identifying the associated proxy for the remote service that is able to  
10 communicate with the remote service using the specified messaging model; and

11 sending the message to the identified proxy for communication to the  
12 remote service using the specified messaging model; and

13 receiving a response to the sent message from the local messaging service  
14 in accordance with the specified messaging model.

1 38. The method of claim 37 including, under control of each of the proxies:  
2 receiving the sent message from the local messaging service; and  
3 sending the message to the remote service in a manner specific to the remote  
4 service.

1           39.     The method of claim 37 wherein each of the proxies are object  
2 instances, and including creating each of the proxies when the proxy is first identified.

1           40.     The method of claim 39 wherein the sending of the messages to the  
2 proxies includes invoking interface methods of the instances.

1           41.     The method of claim 40 wherein at least one of the proxies has distinct  
2 interface methods corresponding to each of the messaging models with which the proxy is  
3 able to communicate, and wherein the sent message to those proxies identifies the specified  
4 messaging model based on selecting the interface method of the proxy that corresponds to  
5 the specified messaging model as the interface method to be invoked.

1           42.     The method of claim 40 wherein the sent messages to the proxies  
2 identify the specified messaging model based on specifying a parameter during the invoking  
3 of the interface method such that the specified parameter corresponds to the specified  
4 messaging model.

1           43.     The method of claim 37 wherein the local messaging service is an  
2 object instance.

1           44.     The method of claim 43 wherein the notifying of the local messaging  
2 service includes invoking an interface method of the local messaging service instance.

1           45.     The method of claim 44 wherein the local messaging service instance  
2 has distinct interface methods corresponding to each of the messaging models with which the  
3 local messaging service is able to communicate, and wherein the specifying of a messaging  
4 model during the notifying of the local messaging service is based on selecting the interface  
5 method of the local messaging service that corresponds to the specified messaging model as  
6 the interface method to be invoked.

1           46.     The method of claim 44 wherein the specifying of a messaging model  
2 during the notifying of the local messaging service is based on specifying a parameter during  
3 the invoking of the interface method such that the specified parameter corresponds to the  
4 specified messaging model.

1           47.     The method of claim 37 wherein the application includes multiple  
2 action handler components each for responding to requests from external clients with  
3 responses generated in a format used by the action handler component, and wherein each  
4 notifying of the local messaging service to send a message is prompted by a request  
5 generated by one of the action handler components.

1           48.     The method of claim 47 wherein the application includes multiple view  
2 handler components each for transforming generated responses in formats used by action  
3 handler components into formats compatible with external clients, and including receiving  
4 responses from remote services and using view handler components to transform the received  
5 responses into formats compatible with external clients.

1           49.     The method of claim 37 wherein at least one of the remote services does  
2 not provide a response that corresponds to the specified messaging model for the  
3 communication, and wherein the identified proxy generates the received response in  
4 accordance with the specified messaging model.

1           50.     The method of claim 37 wherein at least one of the identified proxies  
2 does not provide a response that corresponds to the specified messaging model for the  
3 communication, and wherein the local messaging service generates the received response in  
4 accordance with the specified messaging model.

1           51.     The method of claim 37 wherein a response generated by at least one of  
2 the remote services does not correspond to the specified messaging model, and wherein the

3 identified proxy generates the received response in accordance with the specified messaging  
4 model.

1 52. The method of claim 37 wherein a response generated by at least one of  
2 the identified proxies does not correspond to the specified messaging model, and wherein the  
3 local messaging service generates the received response in accordance with the specified  
4 messaging model.

1 53. A computer-readable medium whose contents cause a computing device  
2 to execute an application able to communicate with remote services using multiple messaging  
3 models, each remote service having for each of the multiple messaging models an associated  
4 proxy able to communicate using that messaging model, by:

5 for each of multiple of the remote services,  
6 notifying a local messaging service to send a message to the remote  
7 service using a specified messaging model;  
8 under control of the local messaging service,  
9 identifying the associated proxy for the remote service that is able to  
10 communicate with the remote service using the specified messaging model; and  
11 sending the message to the identified proxy for communication to the  
12 remote service using the specified messaging model; and  
13 receiving a response to the sent message from the local messaging service  
14 in accordance with the specified messaging model.

1 54. The computer-readable medium of claim 53 wherein the computer-  
2 readable medium is a data transmission medium transmitting a generated data signal  
3 containing the contents.

1 55. The computer-readable medium of claim 53 wherein the computer-  
2 readable medium is a memory of a computer system.

1           56. A computing device for executing an application to communicate with  
2 remote services using multiple messaging models, each remote service having for each of the  
3 multiple messaging models an associated proxy able to communicate using that messaging  
4 model, comprising:

5           an application capable of, for each of multiple of the remote services, notifying  
6 a local messaging service to send a message to the remote service using a specified  
7 messaging model; and

8           the local messaging service capable of identifying the associated proxy for the  
9 remote service that is able to communicate with the remote service using the specified  
10 messaging model, and of sending the message to the identified proxy for communication to  
11 the remote service using the specified messaging model.

1           57. A computing device for executing an application to communicate with  
2 remote services using multiple messaging models, each remote service having for each of the  
3 multiple messaging models an associated proxy able to communicate using that messaging  
4 model, comprising:

5           means for, for each of multiple of the remote services, notifying a local  
6 messaging service to send a message to the remote service using a specified messaging model  
7 and receiving a response to the sent message from the local messaging service in accordance  
8 with the specified messaging model; and

9           means for identifying the associated proxy for the remote service that is able to  
10 communicate with the remote service using the specified messaging model, and for sending  
11 the message to the identified proxy for communication to the remote service using the  
12 specified messaging model.

1           58. A method in a computer system for sending messages to remote  
2 applications using multiple messaging models, each remote application having at least one  
3 associated proxy able to communicate using at least one of the messaging models, the  
4 method comprising:

5           for each of multiple of the remote applications,  
6           receiving a request from a client to send a message to the remote  
7 application using a specified messaging model;  
8           retrieving information associated with the remote application that  
9 identifies at least one proxy associated with the remote application;  
10          when one of the identified proxies is able to communicate using the  
11 specified messaging model,  
12          sending the message to that one proxy for communication to the  
13 remote application using the specified messaging model; and  
14          providing a response to the client in accordance with the sending of  
15 the message using the specified messaging model; and  
16          when none of the identified proxies are able to communicate using the  
17 specified messaging model,  
18          selecting one of the identified proxies that is able to communicate  
19 using a messaging model distinct from the specified messaging model;  
20          sending the message to the selected proxy for communication to the  
21 remote application using the distinct messaging model; and  
22          providing a response to the client in such a manner as to simulate  
23 using the specified messaging model for the sending of the message.

1           59. The method of claim 58 including, under control of each of the proxies:  
2           receiving the sent message; and  
3           sending the message to the remote application in a manner specific to the  
4 remote application.

1           60.     The method of claim 58 wherein each of the proxies are object  
2 instances, and including creating each of the proxies when the proxy is first identified.

1           61.     The method of claim 60 wherein the sending of the messages to the  
2 proxies includes invoking interface methods of the instances.

1           62.     The method of claim 61 wherein at least one of the proxies has distinct  
2 interface methods corresponding to each of the messaging models with which the proxy is  
3 able to communicate, and wherein the sent messages to those proxies identify the specified  
4 messaging model based on selecting the interface method of the proxy that corresponds to  
5 the specified messaging model as the interface method to be invoked.

1           63.     The method of claim 61 wherein the sent messages to the proxies  
2 identify the specified messaging model based on specifying a parameter during the invoking  
3 of the interface method such that the specified parameter corresponds to the specified  
4 messaging model.

1           64.     The method of claim 58 wherein the application includes multiple  
2 action handler components each for responding to requests from external clients with  
3 responses generated in a format used by the action handler component, and wherein the  
4 clients from whom the requests are received are each one of the action handler components.

1           65.     The method of claim 64 wherein the application includes multiple view  
2 handler components each for transforming generated responses in formats used by action  
3 handler components into formats compatible with external clients, and including, after the  
4 providing of each response to one of the action handler components, using one of the view  
5 handler components to transform the received response into a format compatible with one of  
6 the external clients.



1           66. A computer-readable medium whose contents cause a computing device  
2 to send messages to remote applications using multiple messaging models, each remote  
3 application having at least one associated proxy able to communicate using at least one of the  
4 messaging models, by:

5           for each of multiple of the remote applications,  
6           receiving a request from a client to send a message to the remote  
7 application using a specified messaging model;  
8           retrieving information associated with the remote application that  
9 identifies at least one proxy associated with the remote application;  
10          when one of the identified proxies is able to communicate using the  
11 specified messaging model,  
12          sending the message to that one proxy for communication to the  
13 remote application using the specified messaging model; and  
14          providing a response to the client in accordance with the sending of  
15 the message using the specified messaging model; and  
16          when none of the identified proxies are able to communicate using the  
17 specified messaging model,  
18          selecting one of the identified proxies that is able to communicate  
19 using a messaging model distinct from the specified messaging model;  
20          sending the message to the selected proxy for communication to the  
21 remote application using the distinct messaging model; and  
22          providing a response to the client in such a manner as to simulate  
23 using the specified messaging model for the sending of the message.

1           67. A computing device for sending messages to remote applications using  
2 multiple messaging models, each remote application having at least one associated proxy  
3 able to communicate using at least one of the messaging models, comprising:  
4           an first component capable of, for each of multiple of the remote applications,  
5 receiving a request from a client to send a message to the remote application using a

6 specified messaging model, of retrieving information associated with the remote application  
7 that identifies at least one proxy associated with the remote application, of, when one of the  
8 identified proxies is able to communicate using the specified messaging model, sending the  
9 message to that one proxy for communication to the remote application using the specified  
10 messaging model and providing a response to the client in accordance with the sending of the  
11 message using the specified messaging model, and of, when none of the identified proxies  
12 are able to communicate using the specified messaging model, selecting one of the identified  
13 proxies that is able to communicate using a messaging model distinct from the specified  
14 messaging model and sending the message to the selected proxy for communication to the  
15 remote application using the distinct messaging model; and

16 multiple proxy copies each capable of receiving a sent message for  
17 communication to one of the remote applications and communicating the received message to  
18 the one remote application.

1 68. A method in a computer system for components of an executing  
2 application to communicate with remote services using multiple messaging models, each  
3 remote service having at least one associated proxy able to communicate using at least one of  
4 the messaging models, the executing application having a messaging service component for  
5 communicating with the associated proxies, having multiple action handler components each  
6 for responding to requests with responses generated in a format used by the action handler  
7 component, and having multiple view handler components each for transforming generated  
8 responses in formats used by action handler components into formats compatible with  
9 external clients, the method comprising:

10 under control of each of multiple of the action handler components,  
11 receiving a request from a remote client to provide a service; and  
12 providing the requested service by,  
13 determining a need for functionality available from one of the remote  
14 services;  
15 notifying the local messaging component to request the needed  
16 functionality from the one remote service using a specified messaging model;

receiving a response from the one remote service via the local  
messaging component;  
generating a response to the remote client in a format used by the  
action handler component;  
determining a view handler component able to transform the  
generated response from the format used by the action handler component to a format  
compatible with the remote client; and  
providing the generated response to the determined view handler  
component for the transforming;  
under control of the local messaging component,  
receiving notifications from multiple action handler components each to  
request functionality from a remote service using a specified messaging model; and  
for each received notification,  
identifying a proxy associated with the remote service such that the  
identified proxy is able to communicate using a messaging model that is compatible with the  
specified messaging model; and  
sending a request for the functionality to the identified proxy using  
the compatible messaging model for communication to the remote service;  
receiving a response to the sent request from the remote service via  
the identified proxy; and  
notifying the action handler component of the received response; and  
under control of each of multiple of the view handler components,  
receiving responses generated by at least one action handler component in  
a format used by the action handler component; and  
for each received response, transforming the received response into a  
format that is compatible with the remote client for whom the received response was  
generated.

69. A method in a computer system for an executing application to  
communicate with remote services using multiple messaging models, the executing

3 application having a local messaging service able to communicate with each of the remote  
4 services using at least one of the multiple messaging models, the method comprising:  
5           for each of multiple of the remote services,  
6                 receiving an indication to send a specified message to the remote service  
7 using a specified messaging model;  
8                 notifying the local messaging service to send the message to the remote  
9 service using the specified messaging model;  
10                under control of the local messaging service,  
11                 determining whether the remote service supports communication  
12 using a messaging model that is compatible with the specified messaging model;  
13                 when the remote service does support the compatible messaging  
14 model communication,  
15                         sending the message to the remote service using the compatible  
16 messaging model; and  
17                         receiving a response from the remote service in accordance  
18 with the compatible messaging model; and  
19                 when the remote service does not support the compatible messaging  
20 model communication,  
21                         sending the message to the remote service using a messaging  
22 model that is incompatible with the specified messaging model; and  
23                         simulating a response to the sent message in accordance with  
24 the specified messaging model; and  
25                 receiving a response to the sent message from the local messaging service  
26 in accordance with the specified messaging model.